CS 134: Networks

Prof. Yaron Singer Spring 2017

1 Overview

In today's lecture we will be discussing matching markets, or markets defined on networks that involve matching in some way. Examples include:

- Kidney exchanges - matching donors to those who need kidney donations
- Medical residencies - matching applicants to residency programs
- School matching - matching students to public schools; Al Roth won the Nobel Prize in 2012 for proposing a matching algorithm for the NYC public school system
- Dating websites - matching people to each other
- Uber - matching drivers to those requesting rides
- Sponsored search - matching firms to search queries based on auctions

2 Matching Basics

We define matchings on general graphs before defining matchings on bipartite graphs.

Definition. *For a given graph $G = (V, E)$, a **matching** is a set of edges $M \subseteq E$ s.t. the endpoints of the edges in M are all unique. That is, if for some $(u, v) \in E$ we have that $(u, v) \in M$ then $(u, w) \notin M$ for any $(u, w) \in E$.*

We will now define two concepts: maximal and maximum matching. A matching is *maximal* if adding an edge violates the matching, and *maximum* if there are no other matchings in the graph with more edges.

Definition. *A matching M is called **maximal** if for any $e \in E \setminus M$ we have that $M \cup \{e\}$ is not a matching. A matching M is called **maximum** if for any other matching $M' \subseteq E$, we have that $|M'| \leq |M|$.*

2.1 Matching on bipartite graphs

To begin, we will analyze matching markets defined on bipartite graphs; but first, we define the concept of a bipartite graph:

Definition. *A graph $G = (N, E)$ is* ***bipartite*** *iff it can be partitioned into two disjoint subsets of nodes U, V such that (1) $U \cup V = N$ and (2) $\forall e \in E, \exists (u, v) \in U \times V : e = (u, v)$. In words, every node is in one of the two subsets and every edge goes between the two subsets (there are no edges within one of the subsets).*

Moreover, we can think about assignments from nodes in U to nodes in V (that is, assignments have the type $U \to V$). We can also think about a matching assignment as a selection of a subset of the edges in the graph.

2.2 Finding Matches

Given a bipartite graph $G = (V, E)$, it is not difficult to find some matching on the graph. For example, consider the empty matching, which includes no edges, i.e. no vertices are matched. This is a valid matching because no node is an endpoint for more than one edge.

However, we are often interested in more useful matches. In many contexts, such as kidney exchanges, we want to match as many nodes as possible. The brute force approach to finding a maximum matching is exponential time. However, there is a simple polynomial time algorithm for finding a maximal matching. Namely, we iterate through the edges in the graph and add the edge to our matching if adding the edge does not break our matching. This algorithm produces a maximal matching in $O(|E|)$.

2.3 Perfect Matching

Now, we can define what it means for a matching assignment to be perfect:

Definition. *Let $G = (U, V, E)$ be a bipartite graph with subsets U and V. Then when $|U| = |V|$, a* ***perfect matching*** *is an assignment of nodes in U to nodes in V such that (1) $u \in U$ is assigned to $v \in V \iff (u, v) \in E$ and (2) no two nodes in U are assigned to the same $v \in V$.*

Now, how can we easily show that a graph has no perfect matching? To do this, we can utilize the concept of a *constricted set*:

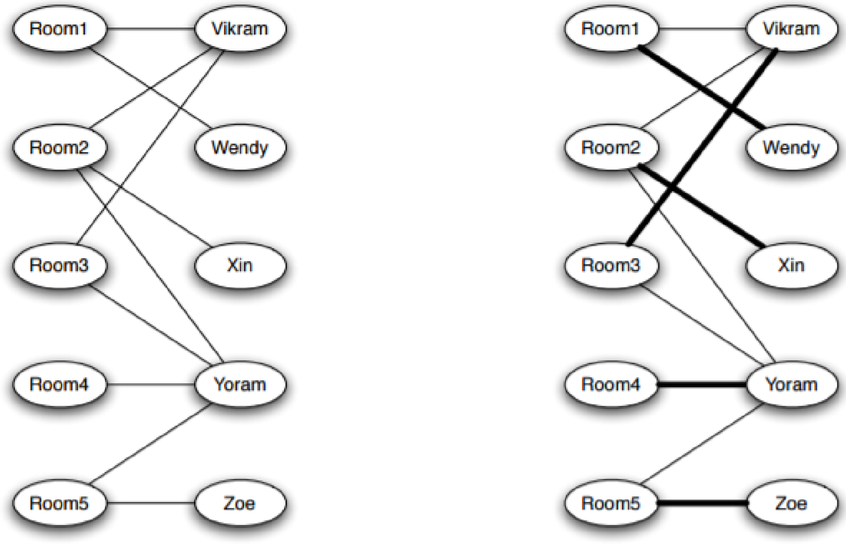
Definition. *A set of vertices $S \subseteq V$ is called a* ***constricted set*** *if $|S| > |N(S)|$ where V is one side of the bipartite graph and $N(S)$ is the neighborhood of S.*

Note that intuitively, a constricted set is a set of nodes on one side of the bipartite graph with fewer edges leaving the set than there are nodes in the set; the existence of a constricted set in a bipartite graph would mean that it is impossible to give a perfect matching for those nodes in the constricted set. This fact is the crux of the matching theorem.

3 The Matching Theorem

Theorem 1 (Matching Theorem)**.** *If a bipartite graph (with $|U| = |V|$) does not have a perfect matching, then it must contain a constricted set of vertices.*

Using this theorem, we can easily show that a graph has no perfect matching by identifying a constricted set of vertices in the bipartite graph. Remarkably, this is the only condition for there not to exist a perfect matching.



(a) *Bipartite Graph* (b) *A Perfect Matching*

Figure 10.1: (a) An example of a bipartite graph. (b) A perfect matching in this graph, indicated via the dark edges.

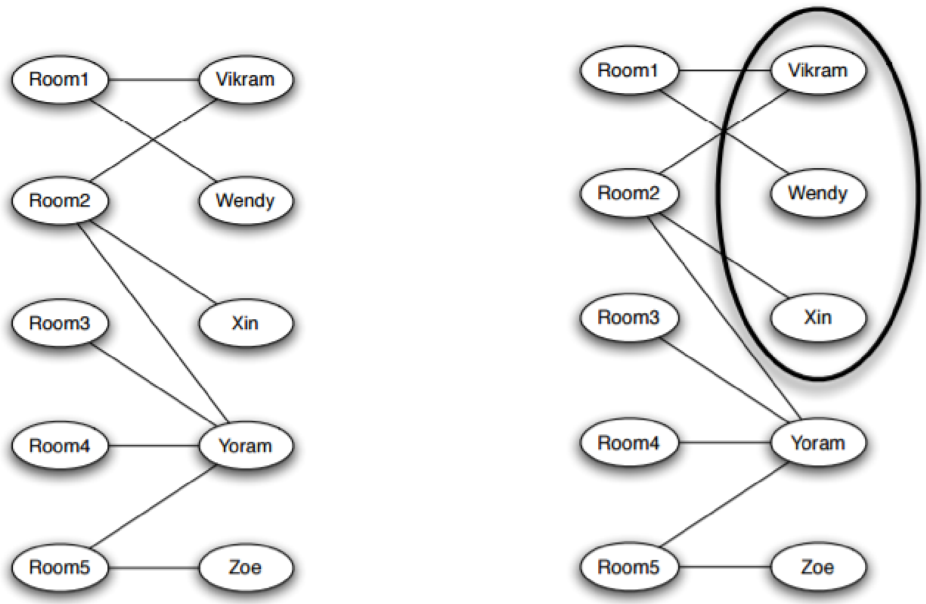
The high-level strategy for proving the matching theorem is to consider a maximum matching in a bipartite graph with no perfect matching. We will trying to expand the matching using *alternating* and *augmenting* paths, and since the process will fail because by assumption we have a maximum matching, we will show we have a constricted set instead.

3.1 Alternating and Augmenting paths

Before proceeding with the proof, we define the following:

> **Definition.** *A path is **alternating** if it is simple (does not repeat any nodes) and alternates between edges in a particular match and not in the match. A path is **augmenting** if it is an alternating path with unmatched endpoints.*

An augmenting path is so named because the existence of an augmenting path means that we can augment, or increase the size of, a matching.. In Figure 10.8a, we have a matching $M = \{(A, X)\}$. Let an edge be a *matching* edges if it is in the matching, and *non-matching* edges otherwise. Possible alternating paths are $\{(W, A), (A, X), (X, B)\}$, $\{(A, X), (X, B)\}$, $\{(A, W)\}$, etc. Note that $p = \{(W, A), (A, X), (X, B)\}$ is an augmenting path (Figure 10.8b) because it is an alternating path that starts and ends with unmatched nodes. We can use this augmenting path to expand M by removing all matching edges in p from M and add in all non-matching edges in p to M, so that our new matching is $M' = \{(W, A), (X, B)\}$ (Figure 10.8c). This operation will always expand the matching because in an augmenting path, the number of non-matching edges is always one more than the number of matching edges. See Figures 10.9 and 10.10 for a matching expansion with an augmenting path in a larger graph.



(a) *Bipartite graph with no perfect matching*

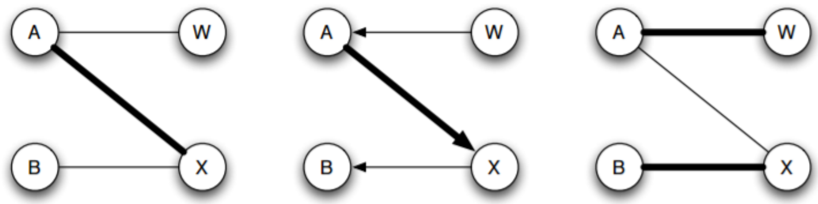
(b) *A constricted set demonstrating there is no perfect matching*

Figure 10.2: (a) A bipartite graph with no perfect matching. (b) A constricted set demonstrating there is no perfect matching.

A natural way to look for an augmenting path is by using a modification of breadth-first search (BFS), which we call AltBFS. Though we omit the details of the algorithm here, intuitively the algorithm works by storing whether or not the most recently traversed edge was matching or nonmatching and only traversing an edge if it is the opposite of what was most recently traversed. If we run AltBFS starting from node w, then we can group the nodes together based on their distance from w as discovered by the search. If w is unmatched, then whenever we encounter an unmatched node in our search, then we have an augmenting path. See figure 10.11 for an example of AltBFS run on on node W in the graph from Figures 10.9 and 10.10.

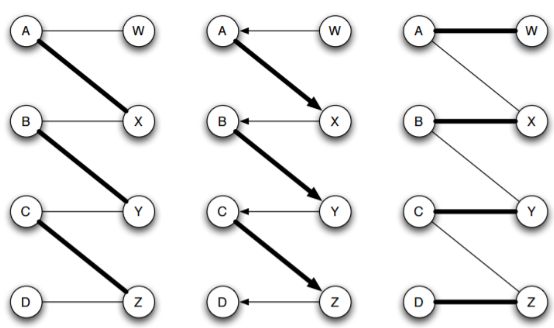
3.2 Proof

We resume the proof by considering what happens when we look for an augmenting path using AltBFS starting from an unmatched node w but fail to find one. Such a search would produce the tree depicted in Figure 10.12. Note that all of the nodes in odd layers come from one partition and all of the nodes in even layer comes from the other position. Furthermore, each odd layer has the same number of nodes as the subsequent even layer since by assumption, we never find an augmenting path so nodes in odd layers must be matched one-to-one with nodes in the following layer. Therefore, there is overall exactly one more node in the even layers than in the odd layers. Finally, all nodes in even layers have their entire neighborhood in the graph because each node's match is in the preceding layer and the rest of its neighborhood either appears in the following



(a) *A matching that is not of maximum size* (b) *An augmenting path* (c) *A larger (perfect) matching*

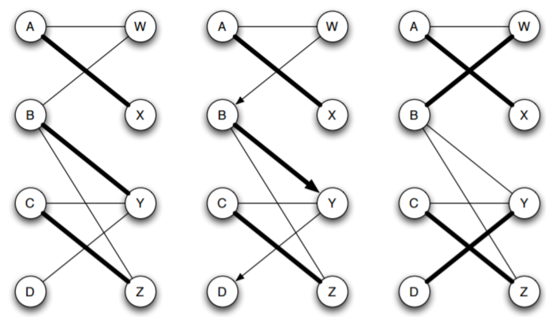
Figure 10.8: (a) A matching that does not have maximum size. (b) What a matching does not have maximum size, we can try to find an *augmenting path* that connects unmatched nodes on opposite sides while alternating between non-matching and matching edges. (c) If we then swap the edges on this path — taking out the matching edges on the path and replacing them with the non-matching edges — then we obtain a larger matching.



(a) *A matching that is not of maximum size* (b) *An augmenting path* (c) *A larger (perfect) matching*

Figure 10.9: The principle used in Figure 10.8 can be applied to larger bipartite graphs as well, sometimes producing long augmenting paths.

(a)



(a) *A matching that is not of maximum size* (b) *An augmenting path* (c) *A larger (perfect) matching*

Figure 10.10: In more complex graphs, finding an augmenting path can require a more careful search, in which choices lead to "dead ends" while others connect two unmatched nodes.

(b)

layer because of the properties of BFS or appeared earlier.

Then overall, the union of the even layers form a constricted set because the size of its neighborhood, i.e. the union of the odd layers, is larger than its size. See Figure 10.13 for an example. Thus, when we search for an augmenting path using AltBFS starting from an unmatched node, then we either succeed or find a constricted set.

We conclude the proof by returning to our original setup: we have a bipartite graph $G = (V, E)$ for which the two partitions are of the same size and for which there is no perfect matching. We consider a maximum matching M and try to enlarge it by searching for an augmented path using the above procedure. We are unable to find an augmenting path because we have a maximum matching, and by the above we must have a constricted set. Thus no perfect matching implies the existence of a constricted set. By contrapositive, if we have no constricted set we must have a perfect matching.

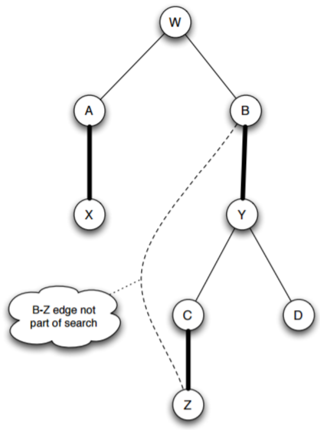


Figure 10.11: In an alternating breadth-first search, one constructs layers that alternately use non-matching and matching edges; if an unmatched node is ever reached, this results in an augmenting path.

(a)

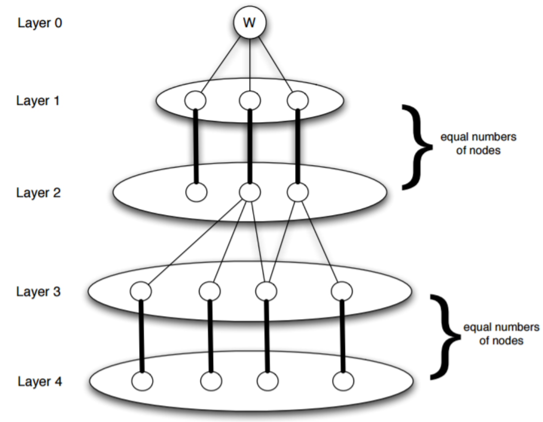
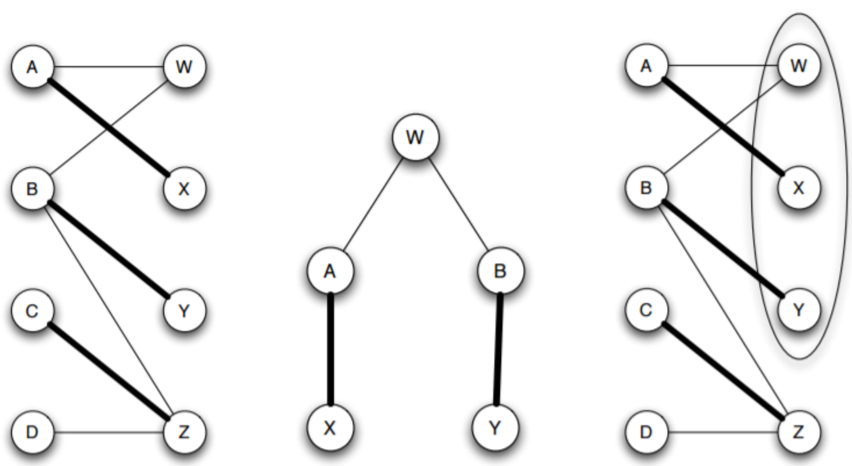


Figure 10.12: A schematic view of alternating breadth-first search, which produces pairs of layers of equal size.

(b)



(a) *A maximum matching that is not perfect*

(b) *A failed search for an augmenting path*

(c) *The resulting constricted set*

Figure 10.13: (a) A matching that has maximum size, but is not perfect. (b) For such a matching, the search for an augment path using alternating breadth-first search will fail. (c) The failure of this search exposes a constricted set: the set of nodes belonging to the even layers.

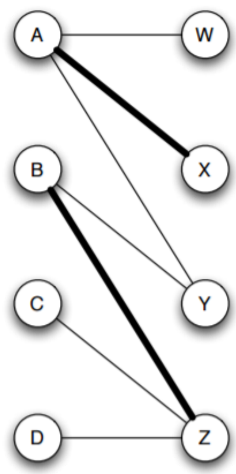


Figure 10.14: If the alternating breadth-first search fails from *any* node on the right-hand side, this is enough to expose a constricted set and hence prove there is no perfect matching. However, it is still possible that an alternating breadth-first search could still succeed from some other node. (In this case, the search from W would fail, but the search from Y would succeed.)